CLAIMS

What is claimed is:

1. A method to analyze a computer program that includes a plurality of blocks of code, the method comprising the steps of:

executing said computer program;

using a counter for tracking each time one of said plurality of blocks of code is executed;

maintaining a counter cache for storing said plurality of counters of said plurality of blocks of code that are most recently executed; and

maintaining a storage area for storing a plurality of counters of said plurality of blocks of code that are not most recently executed.

2. The method of claim 1, further comprising the step of:

identifying when said counter cache is full.

3. The method of claim 2, further comprising the step of:

copying one of said plurality of counters of said plurality of blocks of code from said counter cache to said storage area when said counter cache is full.

4. The method of claim 3, wherein the copying steps further comprises the steps of:

determining which of said plurality of counters of said plurality of blocks of code that are most recently executed is least recently executed; and

copying said least recently executed block of code from said counter cache to said storage area when said counter cache is full.

5. The method of claim 3, further comprising the step of:

checking said code cache to determine if a block of code is being executed for other than the first time; and

loading a counter associated with said block of code being executed for other than the first time, into said counter cache.

6. A system for analyzing a computer program that includes a plurality of blocks of code, comprising:

means for executing said computer program;

means for counting each time one of said plurality of blocks of code is executed;

means for maintaining a counter cache for storing said counting means of said plurality of blocks of code that are most recently executed; and

means for maintaining a storage area for storing said counting means of said plurality of blocks of code that are not most recently executed.

7. The system of claim 6, further comprising:

means for identifying when said counter cache is full.

8. The system of claim 7, further comprising:

means for copying one of said plurality of counting means of said plurality of blocks of code from said counter cache to said storage area when said counter cache is full..

9. The system of claim 8, wherein said identifying means further comprises:

means for determining which of said plurality of counting means of said plurality of blocks of code in said counter cache is least recently executed; and

means for copying said least recently executed block of code from said counter cache to said storage area when said counter cache is full.

10. The system of claim 8, further comprising:

means for checking said code cache to determine if a block of code is being executed for other than the first time; and

means for loading a counting means associated with said block of code being executed for other than the first time, into said counter cache.

11. A computer readable medium for analyzing a computer program that includes a plurality of blocks of code, comprising:

logic for executing said computer program;

logic for counting each time one of said plurality of blocks of code is executed;

logic for storing said counting logic of said plurality of blocks of code that are most recently executed; and

logic for storing said counting logic of said plurality of blocks of code.

12. The computer readable medium of claim 11, further comprising:

logic for identifying when said most recently executed storing logic is full.

13. The computer readable medium of claim 12, further comprising:

logic for copying one of said plurality of counting logic of said plurality of blocks of code from said most recently executed storing logic to said storage logic when said most recently executed storing logic is full.

14. The computer readable medium of claim 13, wherein said logic for identifying further comprises:

logic for determining which of said plurality of counting logic of said plurality of blocks of code in said most recently executed storing logic is least recently executed; and

logic for copying said least recently executed block of code from said most recently executed storing logic to said storage logic when said most recently executed storing logic is full.

15. The computer readable medium of claim 13, wherein said logic for identifying further comprises:

logic for checking said most recently executed storing logic to determine if a block of code is being executed for other than the first time; and

logic for loading a counting means associated with said block of code being executed for other than the first time, into said most recently executed storing logic.

16. A system for analyzing a computer program that includes a plurality of blocks of code, the system comprising:

a counter that tracks each time one of said plurality of blocks of code is executed;

a counter cache that stores said plurality of counters of said plurality of blocks of code that are most recently executed; and

a storage area that stores a plurality of counters of said plurality of blocks of code that are not most recently executed code.

17. The system of claim 16, further comprising:

logic that identifies when said counter cache is full.

18. The system of claim 17, wherein said logic copies one of said plurality of counters of said plurality of blocks of code from said counter cache to said storage area when said counter cache is full.

19. The system of claim 17, wherein said logic determines which of said plurality of counters of said plurality of blocks of code in counter cache is least recently

executed, and copies one of said plurality of counters of said plurality of blocks of code from said counter cache to said storage area when said counter cache is full.

20. The system of claim 17, wherein said logic checks said code cache to determine if a block of code is being executed for other than the first time, and loads a counter associated with said block of code being executed for other than the first time, into said counter cache.